WE CLAIM:

1. Apparatus for processing data, said apparatus comprising:

a processor operable in a plurality of modes and a plurality of domains, said plurality of domains comprising a secure domain or a non-secure domain, said plurality of modes including:

at least one secure mode being a mode in said secure domain; and

at least one non-secure mode being a mode in said non-secure domain;

wherein

when said processor is executing a program in a secure mode said program has access to secure data which is not accessible when said processor is operating in a non-secure mode; and wherein

said processor is responsive to a switching request to initiate a switch between a secure mode and a non-secure mode under control of a mode switching program starting at a location specified by an exception vector associated with said switching request.

2. Apparatus as claimed in claim 1, wherein said switching request is a mode switching software interrupt instruction and said exception vector is a mode switching software interrupt vector.

3. Apparatus as claimed in claim 2, wherein said mode switching software interrupt instruction includes at least one operand operable to control subsequent processing.

4. Apparatus as claimed in claim 1, wherein said switching request is an attempt to change a status flag to switch between a secure mode and a non-secure mode.

5. Apparatus as claimed in claim 4, wherein any attempt by a program to change a status flag to switch between a non-secure mode and a secure mode is one of trapped as an undefined instruction exception with said exception vector being an undefined instruction exception vector or ignored.

6. Apparatus as claimed in claim 1, wherein different exception vectors are associated with said switching request in dependence upon whether said processor is in a secure mode or a non-secure mode.

7. Apparatus as claimed in claim 1, wherein said processor is also operable in a monitor mode and switching between said secure mode and said non-secure mode takes place via said monitor mode, said processor being responsive to said switching request to switch to said monitor mode, said mode switching program being a monitor program operable to at least partially in said monitor mode manage switching between said secure mode and said non-secure mode.

8. Apparatus as claimed in claim 7, wherein said processor includes a register bank and said monitor program is operable to flush at least a portion of said register bank shared between said secure mode and said non-secure mode when switching from said secure mode to said non-secure mode such that no secure data held within said register bank may pass from said secure mode to said non-secure mode other than as permitted by said monitor program.

9. A method of processing data, said method comprising the steps of:

executing a program using a processor operable in a plurality of modes and a plurality of domains, said plurality of domains comprising a secure domain or a non-secure domain, said plurality of modes including:

at least one secure mode being a mode in said secure domain; and

at least one non-secure mode being a mode in said non-secure domain; wherein

when said processor is executing a program in a secure mode said program has access to secure data which is not accessible when said processor is operating in a non-secure mode; and

in responsive to a switching request initiating a switch between a secure mode and a non-secure mode under control of a mode switching program starting at a location specified by an exception vector associated with said switching request.

10. A method as claimed in claim 9, wherein said switching request is a mode switching software interrupt instruction and said exception vector is a mode switching software interrupt vector.

11. A method as claimed in claim 10, wherein said mode switching software interrupt instruction includes at least one operand operable to control subsequent processing.

12. A method as claimed in claim 9, wherein said switching request is an attempt to change a status flag to switch between a secure mode and a non-secure mode.

13. A method as claimed in claim 12, wherein any attempt by a program to change a status flag to switch between a non-secure mode and a secure mode is one of trapped as an undefined instruction exception with said exception vector being an undefined instruction exception vector and ignored.

14. A method as claimed in claim 9, wherein different exception vectors are associated with said switching request in dependence upon whether said processor is in a secure mode or a non-secure mode.

15. A method as claimed in claim 9, wherein said processor is also operable in a monitor mode and switching between said secure mode and said non-secure mode takes place via said monitor mode, said processor being responsive to said switching

request to switch to said monitor mode, said mode switching program being a monitor program operable to at least partially in said monitor mode manage switching between said secure mode and said non-secure mode.

16. A method as claimed in claim 15, wherein said processor includes a register bank and said monitor program is operable to flush at least a portion of said register bank shared between said secure mode and said non-secure mode when switching from said secure mode to said non-secure mode such that no secure data held within said register bank may pass from said secure mode to said non-secure mode other than as permitted by said monitor program.

17. A computer program product having a computer program operable to control a data processing apparatus in accordance with a method as claimed in claim 8.